1. **What are parent properties in flex box.**

In the context of the Flexbox layout model, "parent properties" typically refer to the properties that are applied to the parent container (the element with **display: flex;** or **display: inline-flex;**) to control the layout and alignment of its child elements. These properties help define how the children are distributed within the flex container.

Some of the key parent properties in Flexbox include:

1. **display**: This is the initial property that activates Flexbox on a container. It can have the values **flex** or **inline-flex**, depending on whether you want to create a block-level or inline-level flex container.
2. **flex-direction**: Determines the direction in which the flex items are placed within the flex container. It can be set to values like **row**, **row-reverse**, **column**, or **column-reverse**.
3. **flex-wrap**: Controls whether the flex items should wrap onto a new line if they overflow the container. Values can be **nowrap**, **wrap**, or **wrap-reverse**.
4. **justify-content**: This property defines how the flex items are aligned along the main axis of the flex container. Common values are **flex-start**, **flex-end**, **center**, **space-between**, and **space-around**.
5. **align-items**: Specifies how the flex items are aligned along the cross axis of the container. Options include **flex-start**, **flex-end**, **center**, **baseline**, and **stretch**.
6. **align-content**: Used to align the flex lines (lines of flex items) along the cross axis when there is extra space in the container. This property is particularly relevant when you have multiple rows or columns of flex items. Values include **flex-start**, **flex-end**, **center**, **space-between**, and **space-around**.
7. **flex-flow**: A shorthand property that combines **flex-direction** and **flex-wrap** into a single declaration. For example, **flex-flow: row wrap;** would set the direction to "row" and enable wrapping.
8. **gap** (or **row-gap** and **column-gap**): Specifies the space between flex items within the container, both horizontally and vertically.

These parent properties allow you to control the overall layout and alignment of the flex container and its child elements, providing flexibility and control over the design of your web page or application.

1. **What are child properties in flex box.**

In the Flexbox layout model, child properties, also known as item properties or child-specific properties, are used to control the sizing and alignment of individual flex items within a flex container. These properties apply to each child element within the flex container and allow you to fine-tune the layout and appearance of those items. Here are some key child properties in Flexbox:

1. **flex**: The **flex** property combines three values: **flex-grow**, **flex-shrink**, and **flex-basis**. It determines how a flex item should grow or shrink within the available space. For example, you can use **flex: 1** to make an item grow proportionally, or **flex: 0 0 auto** to prevent it from growing or shrinking.
2. **order**: This property allows you to change the visual order of flex items within the container. Items with a lower **order** value appear first in the order, while items with a higher value appear later.
3. **align-self**: While the **align-items** property controls the alignment of all items along the cross-axis of the container, **align-self** allows you to override the alignment for a specific item. Values include **flex-start**, **flex-end**, **center**, **baseline**, and **stretch**.
4. **flex-grow**: It specifies how much an item should grow relative to the other items when there is extra space in the flex container. A higher value means the item will take up more available space.
5. **flex-shrink**: This property controls how much an item should shrink when there is not enough space in the flex container. A higher value makes the item more likely to shrink.
6. **flex-basis**: Determines the initial size of a flex item before the available space is distributed. It can be set to specific values like **auto**, **content**, or a fixed size.
7. **align-content** (for multi-line flex containers): While primarily a parent property, it can be used on individual flex items to control their alignment within their respective flex lines.

These child properties give you control over how each item within the flex container behaves, allowing you to create flexible and responsive layouts where each item can have its own unique sizing and alignment rules.